

MACHINE ABS_DISPLAY4

REFINES ABS_DISPLAY3

SEES CDIS_CONTEXT4

VARIABLES

database

page_selections

pages

previous_pages

private_pages

trq

time_now

edd_acks_required

concealed_displays /* For all EDDs there is a means of toggling the reveal/ conceal state of the display. This affects only those pages on display that consist of a permanent background and an overlay. Concealed displays models the set of EDDs for which the overlay is concealed. The overlays of pages displayed on all other EDDs are revealed. The act of toggling reveal/ conceal at an EDD adds the EDD to the set if it is not a member beforehand, otherwise removes it. */

INVARIANTS

inv1 : *concealed_displays* \subseteq *EDD_id* /* Conealed displays are a subset of all display positions. */

inv2 : *concealed_displays* \subseteq *dom*(*page_selections*) /* The reveal/ conceal facility applies only to EDDs displaying pages. */

EVENTS

INITIALISATION

BEGIN

act3 : *page_selections* := \emptyset

act8 : *database* : \in *Attr_id* \rightarrow *Attrs*

act9 : *previous_pages* := \emptyset

act10 : *private_pages* := \emptyset

act1 : *pages* := \emptyset

act11 : *trq* := \emptyset

act12 : *time_now* : \in *Date_time*

act4 : *edd_acks_required* := \emptyset

act2 : *concealed_displays* := \emptyset

END

EVENT UPDATE_DATABASE

REFINES UPDATE_DATABASE

ANY

ups

changed_ais

selected_ais

WHERE

grd2 : *ups* \in *Attr_id* \leftrightarrow *Attrs*

```

    grd4 : (ups; last_update) = dom(ups) × {time_now}
    grd3 : changed_ais ⊆ Attr_id
    grd6 : changed_ais = {ai | ai ∈ dom(ups) ∧ value(database(ai)) ≠ value(ups(ai))}
    grd1 : selected_ais ∈ EDD_id ↔ Attr_id
    grd8 : selected_ais = page_selections; pages; contents; fields; attr
  THEN
    act1 : database := database ⧸ ups
    act2 : edd_acks_required := edd_acks_required ∪ ((selected_ais ▷ changed_ais) ▷ Acknowledge)
  END

```

EVENT SET_DATA_VALUE**REFINES** SET_DATA_VALUE

```

  ANY
    ei
    ai
    av
    a
    selected_ais
  WHERE
    grd1 : ei ∈ EDD_id
    grd4 : ei ∈ EDITORS
    grd2 : ai ∈ Attr_id
    grd3 : av ∈ Attr_value
    grd5 : a ∈ Attrs
    grd6 : value(a) = av
    grd7 : last_update(a) = time_now
    grd9 : selected_ais ∈ EDD_id ↔ Attr_id
    grd8 : selected_ais = page_selections; pages; contents; fields; attr
  THEN
    act1 : database(ai) := a
    act2 : edd_acks_required := edd_acks_required ∪ ((selected_ais ▷ {ai}) ▷ Acknowledge)
  END

```

EVENT DISPLAY_PAGE**REFINES** DISPLAY_PAGE

```

  ANY
    ei
    no
  WHERE
    grd1 : ei ∈ EDD_id
    grd2 : no ∈ Page_number
    grd3 : no ∈ dom(pages)
  THEN
    act1 : page_selections(ei) := no
    act2 : edd_acks_required := {ei} ⧸ edd_acks_required
  END

```

EVENT DISMISS_PAGE

/* This event has been affected by the use of concealed_displays. */

REFINES DISMISS_PAGE

```

  ANY
    ei
  WHERE
    grd1 : ei ∈ EDD_id
    grd2 : ei ∈ dom(page_selections)
  THEN
    act2 : page_selections := {ei} ⧸ page_selections
  END

```

$act1 : edd_acks_required := \{ei\} \triangleleft edd_acks_required$
 $act3 : concealed_displays := concealed_displays \setminus \{ei\}$ */* When a page deleted it also should be removed from the concealed_displays, if it is necessary. */*
END

EVENT ADD_PAGE**REFINES ADD_PAGE****ANY***ei**no**pc**p***WHERE** $grd1 : ei \in EDD_id$ $grd2 : no \in Page_number$ $grd3 : pc \in Page_contents$ $grd4 : ei \in EDITORS$ $grd5 : p \in Page$ $grd6 : contents(p) = pc$ $grd7 : creation_date(p) = time_now$ **THEN** $act1 : private_pages(no) := p$ **END****EVENT RELEASE_PAGE****REFINES RELEASE_PAGE****ANY***no**selected_ais**pc1**pc2***WHERE** $grd1 : no \in Page_number$ $grd2 : no \in dom(private_pages)$ $grd3 : no \in dom(pages)$ $grd5 : pc1 = contents(pages(no))$ $grd6 : pc2 = contents(private_pages(no))$ $grd4 : (pc1 \in Overlay_Page_contents \Rightarrow pc2 \in Overlay_Page_contents)$ */* Strengthening**the guard which reflect the new refinement of page-layout */* $grd8 : selected_ais \in EDD_id \leftrightarrow Attr_id$ $grd7 : selected_ais = (page_selections \triangleright \{no\}); private_pages; contents; fields; attr$ **THEN** $act1 : pages(no) := private_pages(no)$ $act3 : previous_pages(no) := pages(no)$ $act2 : private_pages := \{no\} \triangleleft private_pages$ $act5 : edd_acks_required := edd_acks_required \triangleleft ((selected_ais \triangleright Acknowledge) \cap edd_acks_required)$ **END****EVENT RELEASE_PAGE_CURRENT****REFINES RELEASE_PAGE_CURRENT****ANY***no**selected_ais***WHERE** $grd1 : no \in Page_number$ $grd2 : no \in dom(private_pages)$ $grd3 : no \notin dom(pages)$

```

    grd5 : selected_ais ∈ EDD_id ↔ Attr_id
    grd4 : selected_ais = (page_selections ▷ {no}); private_pages; contents; fields; attr
THEN
    act1 : pages(no) := private_pages(no)
    act2 : private_pages := {no} ⋈ private_pages
    act3 : edd_acks_required := edd_acks_required ⋈ ((selected_ais ▷ Acknowledge) ∩ edd_acks_required)
END

```

EVENT RELEASE_PAGE_OVERLAY

/* This event has been affected by the use of concealed_displays. */

REFINES RELEASE_PAGE**ANY**

no
selected_ais
edd_ids

WHERE

grd1 : no ∈ Page_number
grd2 : no ∈ dom(private_pages)
grd3 : no ∈ dom(pages)
grd6 : contents(pages(no)) ∈ Overlay_Page_contents
grd7 : contents(private_pages(no)) ∉ Overlay_Page_contents
grd5 : selected_ais ∈ EDD_id ↔ Attr_id
grd4 : selected_ais = (page_selections ▷ {no}); private_pages; contents; fields; attr
grd8 : edd_ids = dom(page_selections ▷ {no})

THEN

act1 : pages(no) := private_pages(no)
act3 : previous_pages(no) := pages(no)
act2 : private_pages := {no} ⋈ private_pages
act4 : edd_acks_required := edd_acks_required ⋈ ((selected_ais ▷ Acknowledge) ∩ edd_acks_required)
act5 : concealed_displays := concealed_displays \ edd_ids /* The initial state of newly re-

leased page- overlay is revealed. */

END**EVENT RELEASE_PAGES_FROM_TRQ**

/* This event has been affected by the use of concealed_displays. */

REFINES RELEASE_PAGES_FROM_TRQ**ANY**

ss
selected_ais
pp
pn
p_nums
edd_ids

WHERE

grd1 : ss ∈ Page_number ↔ Page
grd3 : pp = {rp | rp ∈ Rel_Page ∧ (release_date(rp) ↦ time_now) ∈ leq}
grd2 : ss = trq ▷ pp
grd5 : selected_ais ∈ EDD_id ↔ Attr_id
grd4 : selected_ais = page_selections; ss; contents; fields; attr
grd6 : pn = dom(ss)
grd7 : p_nums = dom(((pages; contents) ▷ Overlay_Page_contents) ∩ ((ss; contents) ▷ Overlay_Page_contents))
grd8 : edd_ids = dom(page_selections ▷ p_nums)

THEN

act1 : pages := pages ⋈ ss
act3 : previous_pages := previous_pages ⋈ (pn ⋈ pages)
act2 : trq := trq \ ss
act4 : edd_acks_required := edd_acks_required ⋈ ((selected_ais ▷ Acknowledge) ∩ edd_acks_required)

act5 : concealed_displays := concealed_displays \ edd_ids /* The initial state of newly released page- overlay is revealed. */

END

EVENT DELETE_PAGE

/* This event has been affected by the use of concealed_displays. */

REFINES DELETE_PAGE

ANY

ei

no

edd_ids

WHERE

grd1 : ei ∈ EDD_id

grd2 : no ∈ Page_number

grd3 : ei ∈ EDITORS

grd4 : edd_ids = dom(page_selections ▷ {no})

THEN

act1 : pages := {no} ◁ pages

act5 : previous_pages := {no} ◁ previous_pages

act2 : private_pages := {no} ◁ private_pages

act3 : trq := {no} ◁ trq

act4 : page_selections := page_selections ▷ {no}

act6 : edd_acks_required := dom(page_selections ▷ {no}) ◁ edd_acks_required

act7 : concealed_displays := concealed_displays \ edd_ids /* When a page deleted it also should be removed from the concealed_displays, if it is necessary. */

END

EVENT ADD_PAGE_TO_TRQ

REFINES ADD_PAGE_TO_TRQ

ANY

ei

no

pc

p

WHERE

grd1 : ei ∈ EDD_id

grd2 : no ∈ Page_number

grd3 : pc ∈ Page_contents

grd4 : ei ∈ EDITORS

grd5 : p ∈ Rel_Page

grd6 : contents(p) = pc

grd7 : creation_date(p) = time_now

grd8 : (time_now ↦ release_date(p)) ∈ leq

THEN

act1 : trq(no) := p

END

EVENT CLOCK

REFINES CLOCK

ANY

time_next

WHERE

grd1 : time_next ∈ Date_time

grd2 : time_now ↦ time_next ∈ leq

grd3 : time_next ≠ time_now

THEN

act1 : time_now := time_next

END

EVENT KNOWLEDGE_DATA

REFINES KNOWLEDGE_DATA

ANY

ei

WHERE

grd1 : *ei* ∈ *EDD_id*

grd2 : *ei* ∈ *dom(page_selections)*

THEN

act1 : *edd_acks_required* := {*ei*} \triangleleft *edd_acks_required*

END

EVENT VIEW_PAGE

/* This event has been affected by the use of concealed_displays. */

REFINES VIEW_PAGE

ANY

ei

dp

ed

WHERE

grd1 : *ei* ∈ *dom(page_selections)*

grd2 : *dp* ∈ *Disp_params*

grd3 : *ed* ∈ *EDD_display*

grd4 : *dp_data(dp)* = *database*

grd5 : *dp_page(dp)* = *pages(page_selections(ei))*

grd7 : *dp_time(dp)* = *time_now*

grd8 : *dp_acks_required(dp)* = *edd_acks_required*[{*ei*}]

grd9 : *dp_concealed(dp)* = *bool(ei* ∈ *concealed_displays*) /* Revealing or concealing a page

can affect the way should be rendered. */

grd6 : *ed* = *disp_values(dp)*

THEN

skip

END

EVENT TOGGLE_REVEAL_CONCEAL1

/* New event introduced in this level in conjunction with the introduction of concealed_pages variable.

*/

ANY

ei

WHERE

grd11 : *ei* ∈ *EDD_id*

grd12 : *ei* ∈ *concealed_displays*

THEN

act11 : *concealed_displays* := *concealed_displays* \ {*ei*} /* Toggles an EDD display be-

tween a state in which background and overlay information are both displayed and a state in which the overlay (if one is defined) is concealed. */

END

EVENT TOGGLE_REVEAL_CONCEAL2

/* New event introduced in this level in conjunction with the introduction of concealed_pages variable.

*/

ANY

ei

WHERE

grd1 : *ei* ∈ *EDD_id*

grd2 : *ei* ∈ *dom(page_selections)*

```
    grd3 : ei ∉ concealed_displays
    grd4 : contents(pages(page_selections(ei))) ∈ Overlay_Page_contents
THEN
    act1 : concealed_displays := concealed_displays ∪ {ei}  /* Toggles an EDD display between
a state in which background and overlay information are both displayed and a state in which the
overlay (if one is defined) is concealed. */
END

END
```